Classical Algorithms from Quantum and Arthur-Merlin Communication Protocols

Lijie Chen
Massachusetts Institute of Technology, Cambridge, MA, USA
lijieche@mit.edu

Ruosong Wang
Carnegie Mellon University, Pittsburgh, PA, USA
ruosongw@andrew.cmu.edu

Abstract

In recent years, the polynomial method from circuit complexity has been applied to several fundamental problems and obtains the state-of-the-art running times (e.g., R. Williams's $n^3/2^{\Omega(\sqrt{\log n})}$ time algorithm for APSP). As observed in [Alman and Williams, STOC 2017], almost all applications of the polynomial method in algorithm design ultimately rely on certain (probabilistic) low-rank decompositions of the computation matrices corresponding to key subroutines. They suggest that making use of low-rank decompositions directly could lead to more powerful algorithms, as the polynomial method is just one way to derive such a decomposition.

Inspired by their observation, in this paper, we study another way of *systematically constructing low-rank decompositions of matrices* which could be used by algorithms – *communication protocols*. Since their introduction, it is known that various types of communication protocols lead to certain low-rank decompositions (e.g., P protocols/rank, BQP protocols/approximate rank). These are usually interpreted as approaches for proving communication lower bounds, while in this work we explore the other direction.

We have the following two generic algorithmic applications of communication protocols:

- **Quantum Communication Protocols and Deterministic Approximate Counting.** Our first connection is that a fast BQP communication protocol for a function f implies a fast deterministic additive approximate counting algorithm for a related pair counting problem. Applying known BQP communication protocols, we get fast deterministic additive approximate counting algorithms for Count-OV (#OV), Sparse Count-OV and Formula of SYM circuits. In particular, our approximate counting algorithm for #OV runs in near-linear time for all dimensions $d = o(\log^2 n)$. Previously, even no truly-subquadratic time algorithm was known for $d = \omega(\log n)$.
- **Arthur-Merlin Communication Protocols and Faster Satisfying-Pair Algorithms.** Our second connection is that a fast $\mathsf{AM}^{\mathsf{cc}}$ protocol for a function f implies a faster-than-bruteforce algorithm for f-Satisfying-Pair. Using the classical Goldwasser-Sisper AM protocols for approximating set size, we obtain a new algorithm for approximate $\mathsf{Max\text{-}IP}_{n,c\log n}$ in time $n^{2-1/O(\log c)}$, matching the state-of-the-art algorithms in [Chen, CCC 2018].

We also apply our second connection to shed some light on long-standing open problems in communication complexity. We show that if the Longest Common Subsequence (LCS) problem admits a fast (computationally efficient) $\mathsf{AM}^{\mathsf{cc}}$ protocol (polylog(n) complexity), then polynomial-size Formula-SAT admits a $2^{n-n^{1-\delta}}$ time algorithm for any constant $\delta > 0$, which is conjectured to be unlikely by a recent work [Abboud and Bringmann, ICALP 2018]. The same holds even for a fast (computationally efficient) $\mathsf{PH}^{\mathsf{cc}}$ protocol.

2012 ACM Subject Classification Theory of computation → Communication complexity

Keywords and phrases Quantum communication protocols, Arthur-Merlin communication protocols, approximate counting, approximate rank

10th Innovations in Theoretical Computer Science (ITCS 2019).
Editor: Avrim Blum; Article No. 23; pp. 23:1–23:20

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Digital Object Identifier 10.4230/LIPIcs.ITCS.2019.23

Related Version A full version of the paper is available at `https://arxiv.org/abs/1811.07515`.

Acknowledgements The first author is grateful to Josh Alman, Chi-Ning Chou, Mika Göös, and Ryan Williams for helpful discussions during this work. We are grateful to anonymous reviewers for many helpful and inspiring comments on this paper. In particular, we thank one anonymous reviewer for pointing out that Theorem 5 can be improved using the polynomial method.

1 Introduction

Recent works have shown that the polynomial method, a classical technique for proving circuit lower bounds [41, 45], can be useful in designing efficient algorithms [48, 50, 6, 10, 8, 36, 7].

At a very high level, these algorithms proceed as follows: (1) identify a key subroutine of the core algorithm which has a certain low-degree polynomial representation; (2) replace that subroutine by the corresponding polynomials, and reduce the whole problem to a certain *batched evaluation problem of sparse polynomials*; (3) embed that polynomial evaluation problem to *multiplication of two low-rank (rectangular) matrices*, and apply the fast rectangular matrix multiplication algorithm [26].

As [9] point out. In term of step (3), these algorithms are ultimately making use of the fact that the corresponding matrices of some circuits or subroutines have low *probabilistic rank*. [9] suggest that the *probabilistic rank*, or various low-rank decompositions of matrices in general[1], could be more powerful than the polynomial method, and lead to more efficient algorithms, as the polynomial method is just one way to construct them.

It has been noted for a long time that communication protocols are closely related to various notions of rank of matrices. To list a few: deterministic communication complexity is lower bounded by the logarithm of the *rank* of the matrix [37]; quantum communication complexity is lower bounded by the logarithm of the *approximate rank* of the matrix [16, 19]; UPP communication complexity is equivalent to the logarithm of the *sign-rank* of the matrix [40].

These connections are introduced (and usually interpreted) as methods for proving communication complexity lower bounds (see, e.g. the survey by Lee and Shraibman [35]), but they can also be interpreted in the other direction, as a way to *systematically construct low-rank decompositions of matrices*.

In this paper, we explore the connection between different types of communication protocols and low-rank decompositions of matrices and establish several applications in algorithm design. For all these connections, we start with an efficient communication protocol for a problem F, which implies an efficiently constructible low-rank decomposition of the corresponding communication matrix of F, from which we can obtain fast algorithms.

In fact, in our applications of quantum communication protocols, we also consider k-party protocols, and our algorithms rely on the approximate low-rank decomposition of the tensor of the corresponding communication problem. To the best of our knowledge, this is the first time that *approximate tensor rank* is used in algorithm design (approximate rank has been used before, see e.g. [11, 18, 13, 12] and the corresponding related works section).[2]

[1] A low probabilistic rank implies a probabilistic low-rank decomposition of the matrix.

[2] We remark that a concurrent work [52] makes algorithmic use of *non-negative tensor approximate rank* to construct an optimal data structure for the succinct rank problem.

1.1 Quantum Communication Protocols and Deterministic Approximate Counting

Our first result is a generic connection between quantum communication protocols and deterministic approximate counting algorithms.

▶ **Theorem 1.** *(Informal) Let $\mathcal{X}, \mathcal{Y}$ be finite sets and $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a Boolean function. Suppose f has a quantum communication protocol $\mathcal{P}$[3] with complexity $C(\mathcal{P})$ and error ε. Then there is a classical deterministic algorithm $\mathcal{C}$ that receives $A \subseteq \mathcal{X}, B \subseteq \mathcal{Y}$ as input, and outputs a number E such that*

$$\left| \sum_{(x,y)\in A\times B} f(x,y) - E \right| \leq \varepsilon \cdot |A| \cdot |B|.$$

Furthermore, $\mathcal{C}$ runs in $(|A| + |B|) \cdot 2^{O(C(\mathcal{P}))}$ time.

We remark here that there is a simple randomized algorithm running in sub-linear time via random-sampling. Thus the above algorithm is indeed a derandomization of that randomized algorithm.

The above theorem can also be easily generalized to the (number-in-hand) k-party case. See Section 2.5 for the definition of the multiparty quantum communication model.

▶ **Theorem 2.** *(Informal) Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k$ be finite sets and $f : \mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k \to \{0,1\}$ be a Boolean function. Suppose f has a k-party quantum communication protocol $\mathcal{P}$ with complexity $C(\mathcal{P})$ and error ε. Then there is a classical deterministic algorithm $\mathcal{C}$ that receives $X_1 \subseteq \mathcal{X}_1, X_2 \subseteq \mathcal{X}_2, \ldots, X_k \subseteq \mathcal{X}_k$ as input, and outputs a number E such that*

$$\left| \sum_{x_1\in X_1, x_2\in X_2, \ldots, x_k\in X_k} f(x_1, x_2, \ldots, x_k) - E \right| \leq \varepsilon \cdot \prod_{i=1}^{k} |X_i|.$$

Furthermore, $\mathcal{C}$ runs in $(|X_1| + |X_2| + \ldots + |X_k|) \cdot 2^{O(C(\mathcal{P}))}$ time.

Sketching Algorithms

In fact, Theorem 2 implies a stronger *sketching algorithm*. Given subsets $X_1, X_2, \ldots, X_k$, the algorithm first computes a $w = 2^{O(C(\mathcal{P}))}$ size sketch sk_i from each X_i in $O(|X_i| \cdot w)$ time deterministically, and the number E can be computed from these sk_i's in $O(k \cdot w)$ time.

The sketch computed by the algorithm is in fact a vector in $\mathbb{R}^w$, and it satisfies a nice additive property. That is, the sketch of $X_1 \sqcup X_2$ (union as a multi-set) is simply $\mathsf{sk}(X_1) + \mathsf{sk}(X_2)$.

Applying existing quantum communication protocols, we obtain several applications of Theorem 1 and Theorem 2.

1.1.1 Set-Disjointness and Approximate #OV and #k-OV

We first consider the famous Set-Disjointness problem (Alice and Bob get two vectors u and v in $\{0,1\}^d$ correspondingly, and want to determine whether $\langle u, v\rangle = 0$), which has an efficient quantum communication protocol [1] with communication complexity $O(\sqrt{d})$.

[3] We need some technical condition on $\mathcal{P}$, see Corollary 29 for details.

The corresponding count problem for Set-Disjointness is the counting version of the Orthogonal Vectors problem (OV), denoted as $\#\mathsf{OV}_{n,d}$. In this problem, we are given two sets of n vectors $S, T \subseteq \{0,1\}^d$, and the goal is to count the number of pairs $u \in S, v \in T$ such that $\langle u, v\rangle = 0$.

Applying the quantum communication protocol for Set-Disjointness and Theorem 2, we immediately get an algorithm for the approximate version of #OV.

▶ **Theorem 3.** *For any d and any constant $\varepsilon > 0$, $\#\mathsf{OV}_{n,d}$ can be approximated deterministically with additive error $\varepsilon \cdot n^2$ in $n \cdot 2^{O(\sqrt{d})}$ time. In particular, it runs in $n^{1+o(1)}$ time when $d = o(\log^2 n)$.*

Comparison with [22]

[22] gives a *deterministic exact counting* algorithm for $\#\mathsf{OV}_{n,c\log n}$, which runs in $n^{2-O(1/\log c)}$ time. Note that their running time is $n^{2-o(1)}$ when $d = \omega(\log n)$, while our algorithm only achieves an additive approximation, but runs in *near-linear* time for all $d = o(\log^2 n)$.

Another closely related problem, Counting Partial Match, is the problem that given n query strings from $\{0,1,\star\}^d$ ($\star$ is a "don't care") and n strings from $\{0,1\}^d$, and the goal is to count the number of matching string and query pairs.

Using known reductions between Partial Match and OV (see, e.g., Section 2 in [6]), together with the approximate counting algorithm for #OV, we can also solve Counting Partial Match approximately in the same running time.

The approximate counting algorithm for #OV can be easily generalized to solve #k-OV, which is the problem that given k sets of n vectors $X_1, X_2, \ldots, X_k \subseteq \{0,1\}^d$, and count the number of k-tuples $u_1 \in X_1, u_2 \in X_2, \ldots, u_k \in X_k$ such that $\langle u_1, u_2, \ldots, u_k\rangle = 0$.[4]

Applying Theorem 2 and observe that the 2-party Set-Disjointness protocol in [1] can be easily generalized to solve the k-party case (in k-party Set-Disjointness, there are k players getting $u_1, u_2, \ldots, u_k$ respectively, and they want to determine whether $\langle u_1, u_2, \ldots, u_k\rangle = 0$), we obtain the following approximate counting algorithm for #k-OV.

▶ **Theorem 4.** *For any integers k, d and any constant $\varepsilon > 0$, $\#k\text{-}\mathsf{OV}_{n,d}$ can be approximated deterministically with additive error $\varepsilon \cdot n^k$ in $n \cdot 2^{O(k\sqrt{d})}$ time. In particular, it runs in $n^{1+o(1)}$ time when k is a constant and $d = o(\log^2 n)$.*

▶ Remark. We remark that similar algorithms with slightly worse running time ($n \cdot d^{O(\sqrt{d})}$ time for additive approximation to $\#\mathsf{OV}_{n,d}$) can also be derived using the polynomial method. However, we think our new algorithms via quantum communication protocols have the following extra benefits: (1) our algorithm is slightly faster, with a running time of $n \cdot 2^{O(\sqrt{d})}$; (2) our algorithm is derived via a general connection. Once the connection is set up, the algorithm follows in an elegant and black-box way. We hope this general connection could stimulate more applications of quantum communication protocols.

1.1.2 Sparse Set-Disjointness and Approximate Sparse #OV

Next we consider a sparse version of Set-Disjointness, in which Alice and Bob get two sparse vectors $u, v \in \{0,1\}^m_{\le d}$[5], and want to decide whether $\langle u, v\rangle = 0$.

---

[4] the generalized inner product of k vectors, is defined as $\langle u_1, u_2, \ldots, u_k\rangle = \sum_{i=1}^{d} \prod_{j=1}^{k} (u_j)_i$.

[5] We use $\{0,1\}^m_{\le d}$ to denote all Boolean vectors of length m with at most d ones.

Using the famous quantum-walk algorithm for Element Distinctness [14], there is an $O(d^{2/3}\log m)$ communication protocol for sparse Set-Disjointness, which is much better than the $O(\sqrt{m})$ protocol for Set-Disjointness when $m \gg d$.

Applying this protocol and Theorem 1, we can give an algorithm for a sparse version of #OV, denoted as #Sparse-OV$_{n,m,d}$, in which we are given sets $A, B \subseteq \{0,1\}^m_{\leq d}$ of n vectors, and the goal is to count the number of distinct $(a,b) \in A \times B$ such that $\langle a,b\rangle = 0$. Formally, we have:

▶ **Theorem 5.** *For integers n, m, d and any constant $\varepsilon > 0$, #Sparse-OV$_{n,m,d}$ can be approximated deterministically with additive error $\varepsilon \cdot n^2$ in*

$$n \cdot 2^{O(d^{2/3}\log(m))}$$

time. In particular, when $m = \mathrm{poly}(d)$ and $d = o\left(\left(\frac{\log n}{\log\log n}\right)^{1.5}\right)$, it runs in $n^{1+o(1)}$ time.

We remark that it is possible to improve Theorem 5 via the polynomial method. Again, we emphasize that our focus here is to provide direct applications of our general framework, with the hope that it could stimulate more applications of quantum communication protocols in the classical settings.

1.1.3 Approximate Counting for Formula ∘ SYM Circuits

Finally, we apply our algorithm to approximately count solutions (i.e., satisfying assignments) to a class of circuits, for which no non-trivial algorithms were previously known.

A Formula ∘ SYM circuit of size m is a formula with {AND, OR, NOT} basis on m SYM gates[6] at the bottom. Using the quantum query algorithm for Formula Evaluation [15] and the split-and-list technique, we obtain the following deterministic approximate counting algorithm for Formula ∘ SYM circuits:

▶ **Theorem 6.** *For any constant $\varepsilon > 0$, the number of solutions to a Formula ∘ SYM circuit of size m can be approximated deterministically within $\varepsilon \cdot 2^n$ additive error in $2^{O(n^{1/2}m^{1/4+o(1)}\sqrt{\log n+\log m})}$ time. In particular, when $m = n^{2-\delta}$ for some $\delta > 0$, the running time is $2^{o(n)}$.*

Previously, even no non-trivial deterministic approximate counting algorithms for AND ∘ SYM circuits were known. A recent line of works [31, 32, 44], culminating in [39], construct a PRG for AND$_m$ ∘ THR circuits with seed length $\mathrm{poly}(\log m, \delta^{-1}) \cdot \log n$, using which one can obtain a quasi-polynomial time deterministic approximate counting algorithm for polynomial size AND ∘ THR circuits. However, their PRG constructions rely on the fact that the solution set of an AND$_m$ ∘ THR circuit is a *polytope*, while the solution set of an AND ∘ SYM circuit may not have such a nice geometric structure.

In fact, the only property we need for SYM gates is that they admit an efficient *classical* k-party communication protocol when the inputs are divided to k players (each player sends the contribution of her part). Our algorithm actually works for the following more general problem.

▶ **Problem 1.** *Given k sets of n vectors $X_1, X_2, \ldots, X_k \subseteq \{0,\ldots,r\}^d$ and d functions $f_1, f_2, \ldots, f_d$ where each f_i is from $[r]^k$ to $\{0,1\}$, and a Boolean formula $\mathcal{F} : \{0,1\}^d \to \{0,1\}$ of $O(1)$ fan-in. Count the number of k-tuples $u_1 \in X_1, u_2 \in X_2, \ldots, u_k \in X_k$ such that*

$$\mathcal{F}(f_1(u_{1,1}, u_{2,1}, \ldots, u_{k,1}), f_2(u_{1,2}, u_{2,2}, \ldots, u_{k,2}), \ldots, f_d(u_{1,d}, u_{2,d}, \ldots, u_{k,d})) = 1.$$

[6] A SYM gate is a gate whose output only depends on the number of ones in the input.

▶ **Theorem 7.** *For any constant $\varepsilon > 0$, the above problem can be solved deterministically in $n \cdot 2^{O(d^{1/2+o(1)} \cdot k(\log d + \log r))}$ time, within $\varepsilon \cdot n^k$ additive error.*

1.2 Arthur-Merlin Communication Protocols and a New Approximate Max-IP Algorithm

Our second connection is an algorithmic application of $\mathsf{AM}^{\mathsf{cc}}$ protocols. We first define $\mathsf{AM}^{\mathsf{cc}}$ protocols formally.

▶ **Definition 8.** An Arthur-Merlin communication protocol ($\mathsf{AM}^{\mathsf{cc}}$) Π for a partial function $F : \mathcal{X} \times \mathcal{Y} \to \{0, 1, \perp\}$[7] proceeds as follows:

- Alice holds input $x \in \mathcal{X}$ and Bob holds input $y \in \mathcal{Y}$.
- Alice and Bob toss some public coins jointly and send the random string $r \in \{0,1\}^*$ to Merlin (r is called the random challenge).
- Based on x, y and the random challenge r, Merlin sends Alice and Bob a proof ψ, and Alice and Bob decide to accept or not independently and deterministically. We require the following conditions:
  - If $F(x, y) = 1$, with probability $1 - \varepsilon$ over the random challenge r, there is a proof ψ from Merlin such that Alice and Bob both accept.
  - If $F(x, y) = 0$, with probability $1 - \varepsilon$ over the random challenge r, there is no proof ψ from Merlin such that Alice and Bob both accept.

We call the parameter ε the error of the protocol Π. Moreover, we say the protocol is *computationally efficient* if Alice and Bob's behavior can be computed in polynomial-time w.r.t. their input lengths.

We show that for any function F, a low-complexity and computationally efficient $\mathsf{AM}^{\mathsf{cc}}$ protocol implies a faster algorithm for the corresponding F-Satisfying-Pair problem (defined below).

For a partial function $F : \mathcal{X} \times \mathcal{Y} \to \{0, 1, \perp\}$, where $\mathcal{X}$ and $\mathcal{Y}$ are two sets, we define F-Satisfying-Pair$_n$ as the problem that given two sets $A \subseteq \mathcal{X}$ and $B \subseteq \mathcal{Y}$ of size n, distinguish between the following two cases: (1) There is an $(x, y) \in A \times B$ such that $F(x, y) = 1$. (2) For all $(x, y) \in A \times B$, $F(x, y) = 0$.

▶ **Theorem 9** (Algorithms from $\mathsf{AM}^{\mathsf{cc}}$ protocols)**.** *Let $F : \mathcal{X} \times \mathcal{Y} \to \{0, 1, \perp\}$ be a partial function. Suppose there is a computationally efficient $\mathsf{AM}^{\mathsf{cc}}$ protocol for F with communication complexity T and error ε. Then for n such that $2^T \leq (\sqrt{\varepsilon} n)^{0.1}$, there is an $O\left(\varepsilon n^2 \cdot \mathrm{polylog}(n) + n \cdot 2^T\right)$ time randomized algorithm for F-Satisfying-Pair$_n$.*

1.2.1 A New Algorithm for Approximate Max-IP

The first application of Theorem 9 is a new algorithm for approximate Maximum Inner Product. We use Max-IP$_{n,d}$ to denote the problem that given sets $A, B \subseteq \{0,1\}^d$ with size n, compute $\mathsf{Max}(A, B) := \max_{(a,b) \in A \times B} \langle a \cdot b \rangle$.

To phrase this as an F-Satisfying-Pair problem, we first define the following gap inner product problem.

[7] $F(x, y) = \perp$ means $F(x, y)$ is undefined.

▶ **Definition 10** (Multiplicative-Gap Inner Product). Consider the following problem, denoted as Gap-Inner-Product$_d$, Alice and Bob hold strings $x, y \in \{0,1\}^d$ respectively, and they are given an integer τ. They want to distinguish between the following two cases: (Yes) $x \cdot y \geq 2\tau$; (No) $x \cdot y \leq \tau$.

Adapting the classical Goldwasser-Sisper AM protocol for approximating set size [28], we can derive an efficient AM^{cc} protocol for Gap-Inner-Product$_d$.

▶ **Lemma 11** (AM^{cc} protocol for Gap-Inner-Product$_d$). *There is an* AM^{cc} *protocol which solves* Gap-Inner-Product$_d$ *with error* ε *and communication complexity* $\log \binom{d}{\leq O(\log \varepsilon^{-1})}$.[8]

Applying Theorem 9, the following algorithm for approximating Max-IP follows directly, matching the previous best algorithm in [23].

▶ **Corollary 12.** *There is an algorithm for computing a* 2*-approximation to* Max-IP$_{n,c\log n}$*, which runs in* $n^{2-1/O(\log c)}$ *time.*

▶ Remark. The constant 2 in Corollary 12 can be replaced by any other constant $\kappa > 1$.

We remark here that a direct application of the Goldwasser-Sisper protocol and parallel repetition leads to a communication protocol with communication complexity $O(\log d \log \varepsilon^{-1})$, which is slightly worse than Lemma 11. In particular, such a protocol only gives an algorithm with running time $n^{2-1/O(\log d)}$, which is worse than $n^{2-1/O(\log c)}$ when $c \ll d = c \log n$. In order to get the improved complexity in Lemma 11, we make use of a clever sampling scheme using Poisson distributions, see Section 4.1 for details.

1.2.2 Evidences that Longest Common Subsequence and Edit Distance do not Have Fast AM^{cc} Protocols

It has been a long-standing open problem in communication complexity to prove an $\omega(\log n)$ AM^{cc} lower bound for any explicit function [17, 29, 30]—it is consistent with our current knowledge that all known natural communication problems have $O(\log n)$ AM^{cc} protocols.

We consider two natural communication problems here, LCS$_d^{cc}$ and Edit-Dist$_d^{cc}$, in which Alice and Bob hold strings $x, y \in \{0,1\}^d$ respectively, and are given an integer τ. Their goal is to decide whether $\mathsf{LCS}(x,y) \geq \tau$ ($\mathsf{Edit\text{-}Distance}(x,y) \geq \tau$).

Our Theorem 9 shows that if LCS^{cc} or Edit-Dist^{cc} admit low-complexity and computationally efficient AM^{cc} protocols, it would imply non-trivial algorithms for the corresponding F-Satisfying-Pair problem. By a known reduction in [3], that would, in turn, implies non-trivial algorithms for Formula-SAT[9]—much faster than the current state-of-the-art [47]! Therefore, at least for these two problems, constructing low-complexity AM^{cc} protocol could be hard, which may also be viewed as an evidence that they do not have efficient AM^{cc} protocols.

▶ **Theorem 13.** *If* LCS$_d^{cc}$ *admits computationally efficient* AM^{cc} *protocols with complexity* $\mathrm{polylog}(d)$*, then Formula-SAT of polynomial-size formulas admits an* $2^{n-n^{1-\delta}}$ *time algorithm for any constant* $\delta > 0$*. The same holds for* Edit-Dist^{cc} *in place of* LCS^{cc}*.*

The state-of-the-art algorithm for Formula-SAT runs in $o(2^n)$ time only when the formula size is smaller than n^3 [47]. It is even purposed as a hypothesis that no $2^n/n^{\omega(1)}$ time algorithm exists for $n^{3+\Omega(1)}$-size Formula-SAT in [2]. Therefore, our results imply that if LCS^{cc} or Edit-Dist^{cc} admits fast (computationally efficient) AM^{cc} protocols, then that would refute the hypothesis in [2]:

[8] $\binom{n}{\leq m}$ denotes $\sum_{i=0}^{m} \binom{n}{i}$.
[9] Formula-SAT is the problem that deciding whether a given formula is satisfiable.

▶ **Corollary 14.** *Under the following hypothesis*[10]*, $\mathit{LCS}_d^{\mathsf{cc}}$ and $\mathit{Edit\text{-}Dist}_d^{\mathsf{cc}}$ do not admit computationally efficient $\mathsf{AM}^{\mathsf{cc}}$ protocols with complexity* $\mathrm{polylog}(d)$*:*

- *There is a constant $\delta > 0$ such that Formula-SAT of polynomial-size formulas requires $2^{n-n^{1-\delta}}$ time.*

In the full version of this paper, we show that the above corollary can be generalized to hold for computationally efficient $\mathsf{PH}^{\mathsf{cc}}$ protocols (see the full version for a formal definition). Formally, we have:

▶ **Theorem 15.** *Under the same hypothesis as in Corollary 14, $\mathit{LCS}_d^{\mathsf{cc}}$ and $\mathit{Edit\text{-}Dist}_d^{\mathsf{cc}}$ do not admit computationally efficient $\mathsf{PH}^{\mathsf{cc}}$ protocols with complexity* $\mathrm{polylog}(d)$*.*

1.3 Related Works

1.3.1 Communication Protocols and Fine-Grained Complexity

Recently, since the breakthrough work of [5], communication protocols have been applied to *fine-grained complexity*, and several tight conditional hardness results are proved for many fundamental approximate problems in P [5, 33, 4, 23, 24, 25, 43].

Among these works, the most related one is [23], in which the author also makes use of the $\mathsf{BQP}^{\mathsf{cc}}$ protocol for Set-Disjointness for a different purpose. In [23], the $\mathsf{BQP}^{\mathsf{cc}}$ protocol is used to established a *reduction* from OV to approximate $\{-1,1\}$-Max-IP[11], thereby showing the SETH-hardness of approximating $\{-1,1\}$-Max-IP. On the other hand, in this work we use $\mathsf{BQP}^{\mathsf{cc}}$ protocols directly for algorithmic purposes.

1.3.2 Other Algorithmic Applications of Approximate Rank

Alon studies the approximate rank of the identity matrix I_n in [11]. It is shown that it is at least $\Omega\left(\frac{\log n}{\varepsilon^2 \log(1/\varepsilon)}\right)$ and at most $O\left(\frac{\log n}{\varepsilon^2}\right)$. Built upon this result, several applications in geometry, coding theory, extremal finite set theory and the study of sample spaces supporting nearly independent random variables are derived. The lower bound also has applications in combinatorial geometry and in the study of locally correctable codes over real and complex numbers, as shown in [18]. In [13, 12], several bounds on approximate rank are derived, together with applications of approximate rank in approximating Nash Equilibria, approximating densest bipartite subgraph and covering convex bodies.

2 Preliminaries

2.1 Fast Rectangular Matrix Multiplication

Similar to previous algorithms using the polynomial method (see, e.g., [50, 10, 6]), our algorithms also make use of algorithms for fast rectangular matrix multiplication.

▶ **Theorem 16** ([26, 27])**.** *There is an $N^2 \cdot \mathrm{polylog}(N)$ time algorithm for multiplying two matrices A and B with size $N \times N^\alpha$ and $N^\alpha \times N$, where $\alpha > 0.172$.*

[10] which is much weaker than the hypothesis in [2]

[11] a variant of Max-IP with vectors in $\{-1,1\}^d$ instead of $\{0,1\}^d$

2.2 Random Variables and Poisson Distributions

Throughout the paper, we use $X \simeq Y$ to mean that X and Y have the same distribution. We use $X \succeq Y$ to denote stochastic dominance, i.e., $X \succeq Y$ iff for any $t \in \mathbb{R}$, $\Pr[X \geq t] \geq \Pr[Y \geq t]$.

We use $\mathrm{Pois}(\lambda)$ to denote a Poisson distribution with parameter λ. We will need the following two facts about Poisson distributions. The proof can be found in the full version.

▶ **Lemma 17.** *Suppose $\{X_i\}_{i=1}^n$ is a set of independent random variables with $X_i \sim \mathrm{Pois}(\lambda_i)$, then $\sum_{i=1}^n X_i \sim \mathrm{Pois}\left(\sum_{i=1}^n \lambda_i\right)$.*

▶ **Lemma 18.** $\Pr\left[\mathrm{Pois}(\lambda) \geq 1.2\lambda\right] \leq e^{-0.01\lambda}$ *and* $\Pr\left[\mathrm{Pois}(\lambda) \leq 0.8\lambda\right] \leq e^{-0.01\lambda}$.

2.3 Tensor Ranks

In this paper we are interested in the approximate tensor rank with respect to the ℓ_∞ norm. For more on approximate tensor rank with respect to other norms and their applications, see [46] and the references therein. Now we introduce some relevant definitions.

▶ **Definition 19.** We say a tensor $T \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_k}$ is *simple* if $T = v_1 \otimes v_2 \otimes \ldots \otimes v_k$ where $v_i \in \mathbb{R}^{n_i}$.

▶ **Definition 20.** For a tensor $T \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_k}$, its $\mathrm{rank}(T)$ is defined to be the smallest integer r such that $T = \sum_{i=1}^r A_r$ and A_i is simple for all $i \in [r]$.

▶ **Definition 21.** For a tensor $T \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_k}$, the approximate rank of T is defined as follows: $\mathrm{rank}_\varepsilon(T) = \min\{\mathrm{rank}(S) \mid \|T - S\|_\infty \leq \varepsilon\}$. Here $\|\cdot\|_\infty$ is the entry-wise ℓ_∞-norm of a tensor.

2.4 Quantum Query Complexity

In this section we recall some previous results on quantum query complexity. Here we emphasize the number of qubits used by the algorithms, which will be crucial when simulating them using classical algorithms.

▶ **Definition 22.** In the Formula Evaluation problem, we are given a formula $\mathcal{F}$ with $\{\mathsf{AND}, \mathsf{OR}, \mathsf{NOT}\}$ basis and $O(1)$ fan-in on n variables $x_1, x_2, \ldots, x_n$. In each query, the algorithm gets the value of x_i, where $i \in [n]$ is determined by the algorithm. The goal is to evaluate the formula.

▶ **Theorem 23** ([15])**.** *The* Formula Evaluation *problem can be solved in $O(n^{1/2+o(1)})$ queries using $O(\mathrm{polylog}(n))$ qubits, with failure probability at most $1/3$.*

▶ Remark. There is an optimal $O(n^{1/2})$ query algorithm for Formula Evaluation [42]. However, that query algorithm doesn't fit in our applications here for two reasons: (1) the algorithm needs $O(n)$ qubits, which is too much for classical simulation; (2) the algorithm is not *computationally efficient* and it takes too much time to compute the corresponding unitary transformation.

▶ **Definition 24.** In the Element Distinctness problem, we are given n elements $X = (x_1, x_2, ..., x_n) \in [m]^n$. In each query, the algorithm gets the value of x_i, where $i \in [n]$ is determined by the algorithm. The goal is to decide whether there are two distinct indices $i \neq j$ such that $x_i = x_j$.

▶ **Theorem 25** ([14])**.** *The* Element Distinctness *problem can be solved in $O(n^{2/3})$ queries using $O(n^{2/3} \log m)$ qubits, with failure probability at most $1/3$.*

2.5 Multiparty Quantum Communication Protocols

In this section, we give our definition of multiparty quantum communication protocols.

Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k$ be finite sets and $f : \mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k \to \{0,1\}$ be a function. In a k-part quantum communication protocols, there are k players $P_1, P_2, \ldots, P_k$, together with a Hilbert space $H = H_1 \otimes H_2 \otimes \ldots \otimes H_k \otimes \overline{H}$. Here H_i serves as the inner working space for player P_i, and $\overline{H}$ is the communication channel between all the players. Each player P_i receives an input $x_i \in \mathcal{X}_i$ and the goal is to determine $f(x_1, x_2, \ldots, x_k)$.

Now we give the formal definition of a k-party quantum communication protocol.

▶ **Definition 26.** A k-part quantum communication protocol $\mathcal{P} = \mathcal{P}(x_1, x_2, \ldots, x_k)$ is a sequence of r unitary transforms $\mathcal{P} = (U_1^{p_1}(x_{p_1}), U_2^{p_2}(x_{p_2}), \ldots, U_r^{p_r}(x_{p_r}))$, such that:

- $U_i^{p_i}(x_{p_i})$ is a unitary transform *acting on* $H_{p_i} \otimes \overline{H_i}$ where $\overline{H_i}$ is a subspace spanned by some qubits of $\overline{H}$[12]. That is, it is the action of p_i-th player P_{p_i}, who is in charge of the i-th turn.
- The sequence $p_1, p_2, \ldots, p_r$, and $\overline{H_1}, \overline{H_2}, \ldots, \overline{H_r}$ are fixed and do not depend on $x_1, \ldots, x_k$. In other words, $\overline{H_i}$ corresponds to the qubits in the channel $\overline{H}$ that player P_{p_i} will modify during its action in the i-th turn, and all players take actions in a fixed, predefined order.
- The communication complexity of $\mathcal{P}$ is defined to be $C(\mathcal{P}) = \sum_{i=1}^{r} \log(\dim(\overline{H_i}))$. The space complexity of P_i is defined to be $S_i(\mathcal{P}) = \log(\dim(H_i \otimes \overline{H}))$.

For a protocol $\mathcal{P} = (U_1^{p_1}(x_{p_1}), U_2^{p_2}(x_{p_2}), \ldots, U_r^{p_r}(x_{p_r}))$, we say $\mathcal{P}$ computes f with error ε if we measure the *first* qubit in $\overline{H}$ on the state $U_r^{p_r}(x_{p_r}) \cdot U_{r-1}^{p_{r-1}}(x_{p_{r-1}}) \cdot \ldots \cdot U_2^{p_2}(x_{p_2}) \cdot U_1^{p_1}(x_{p_1}) \cdot |0\rangle$, we get $f(x_1, x_2, \ldots, x_k)$ with probability at least $1 - \varepsilon$, for all $x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2, \ldots, x_k \in \mathcal{X}_k$.

▶ Remark. We remark that our definition here is more complicated than the usual definition of quantum communication protocols in the literature (see, e.g., [34]), but nonetheless, it is equivalent to them. We choose to formulate it in such a way because it is easier to describe the classical simulation of quantum communication protocols for low approximate rank decompositions, and the simulation of quantum query algorithms (see below).

2.5.1 Simulating Quantum Query Algorithm in Quantum Communication Protocols

Quantum communication protocols can be built upon quantum query algorithms (see, e.g., [20]). Here we give an example to show how to simulate a quantum query algorithm for Formula Evaluation to construct a quantum communication protocol for the communication problem corresponding to Problem 1, under our definition.

In the corresponding k-party communication problem, there are k players, and the i-th player P_i is given a vector $u_i \in [r]^d$. There are d functions $f_1, f_2, \ldots, f_d$ where each f_i is from $[r]^k$ to $\{0,1\}$, and a Boolean formula $\mathcal{F} : \{0,1\}^d \to \{0,1\}$ of $O(1)$ fan-in. Set $v(i) = f_i(u_{1,i}, u_{2,i}, \ldots, u_{k,i})$. Their goal is to compute $\mathcal{F}(v(1), v(2), \ldots, v(d))$.

Now we show how to construct a quantum communication protocol for the above problem.

▶ **Example 27.** Assume that the first player runs a quantum query algorithm for the Formula Evaluation problem. For the simulation, we only need to implement the following query gate O_v: $|i\rangle |b\rangle \to |i\rangle |b \oplus v(i)\rangle$, where i is the index of a variable written in binary form and $v(i)$ is the corresponding input bit to $\mathcal{F}$.

[12] i.e., $U_i^{p_i}(x_{p_i})$ does not alter qubits other than those in $H_{p_i} \otimes \overline{H_i}$.

We first specify the channel, $\overline{H}$ is defined as $\overline{H}_{\mathsf{index}} \otimes \overline{H}_{\mathsf{output}} \otimes \overline{H}_1 \otimes \cdots \otimes \overline{H}_k$. $\overline{H}_{\mathsf{index}}$ and $\overline{H}_{\mathsf{output}}$ together simulate the query gate, and $\overline{H}_i$ is the place for player P_i to write her number.

In the beginning, all qubits in $\overline{H}$ are $|0\rangle$. When the first player wants to apply O_v on some qubits in H_1, it first swaps the qubits containing i and b in H_1 with $\overline{H}_{\mathsf{index}}$ and $\overline{H}_{\mathsf{output}}$ in $\overline{H}$.

Each player P_j in turn reads i in $\overline{H}_{\mathsf{index}}$ and writes the value of $u_{j,i}$ to qubits in $\overline{H}_j$. Note that each player can write the value of $u_{j,i}$ to qubits in $\overline{H}_j$ using a unitary transformation since all qubits in $\overline{H}_j$ are $|0\rangle$ at the beginning, by assumption.

Now, given the value of i and $u_{1,i}, u_{2,i}, \ldots, u_{k,i}$, the first player maps $|i\rangle |b\rangle$ to $|i\rangle |b \oplus v(i)\rangle$ via a unitary transformation. Now the gate O_v is implemented, but we still have to clean up the garbages in $\overline{H}_j$'s, and set them back to $|0\rangle$'s. This can be done by applying reverse transforms of all applied unitary transformation, in the reverse order.

The communication complexity of this protocol is $O(Q \cdot k(\log d + \log r))$, where Q is the query complexity of the quantum query algorithm. Also, using the algorithm in Theorem 23, the communication complexity of this protocol is $O(n^{1/2+o(1)} \cdot k(\log d + \log r))$.

3 Approximate Counting Algorithms from Quantum Communication Protocols

Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k$ be finite sets and $f : \mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k \to \{0,1\}$ be a function. Let $M_f \in \{0,1\}^{|\mathcal{X}_1| \times |\mathcal{X}_2| \times \ldots \times |\mathcal{X}_k|}$ denote the Boolean tensor whose $(x_1, x_2, \ldots, x_k)$ entry is $f(x_1, x_2, \ldots, x_k)$. The following connection between 2-party quantum communication complexity and approximate rank is first observed in [21]. This result can be generalized to the k-party case to get the following theorem. Full details can be found in the full version of this paper.

▶ **Theorem 28.** *Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k$ be finite sets and $f : \mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k \to \{0,1\}$ be a Boolean function. Suppose there exists a k-party efficient quantum communication protocol $\mathcal{P}$, such that $\mathcal{P}$ gives the correct answer with probability at least $1-\varepsilon$ on every input, then $\mathrm{rank}_\varepsilon(M_f) \leq 2^{O(C(\mathcal{P}))}$, or equivalently, there exist simple tensors $A_1, A_2, \ldots, A_{2^{O(C(\mathcal{P}))}}$ such that*

$$\left\| M_f - \sum_{i=1}^{2^{O(C(\mathcal{P}))}} A_i \right\|_\infty \leq \varepsilon.$$

In the full version of this paper, we further show how to use classical deterministic algorithms to simulate quantum communication protocols. Notice that here the time complexity depends on the space complexity of the quantum communication protocol to use.

▶ **Corollary 29.** *Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k$ be finite sets and $f : \mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k \to \{0,1\}$ be a Boolean function. Suppose there exists a k-party efficient quantum communication protocol $\mathcal{P}$, such that $\mathcal{P}$ gives the correct answer with probability at least $1-\varepsilon$ on every input, and all the unitary transformation used in the $\mathcal{P}$ can be constructed in polynomial time (with respect to their sizes) by a deterministic classical algorithm. Then there exists k deterministic classical algorithms $\mathcal{A}_{\mathcal{X}_1}, \mathcal{A}_{\mathcal{X}_2}, \ldots, \mathcal{A}_{\mathcal{X}_k}$ such that $\mathcal{A}_{\mathcal{X}_i}$ runs in $2^{O(C(\mathcal{P})+S_i(\mathcal{P}))}$ time, receives $x_i \in \mathcal{X}_i$ as input and outputs a vector $\mathcal{A}_{\mathcal{X}_i}(x_i) \in \mathbb{R}^{2^{O(C(\mathcal{P}))}}$, and for any $x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2, \ldots, x_k \in \mathcal{X}_k$,*

$$-\varepsilon \leq \langle \mathcal{A}_{\mathcal{X}_1}(x_1), \mathcal{A}_{\mathcal{X}_2}(x_2), \ldots, \mathcal{A}_{\mathcal{X}_k}(x_k) \rangle - f(x_1, x_2, \ldots, x_k) \leq \varepsilon.$$

Based on Corollary 29, for any Boolean function $f : \mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k \to \{0,1\}$ with an efficient efficient quantum communication protocol, there also exists an efficient approximate counting algorithm for f.

▶ **Theorem 30.** *Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k$ be finite sets and $f : \mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k \to \{0,1\}$ be a Boolean function. Suppose there exists a k-party efficient quantum communication protocol $\mathcal{P}$, such that $\mathcal{P}$ gives the correct answer with probability at least $1-\varepsilon$ on every input, and all the unitary transformation used in the $\mathcal{P}$ can be constructed in polynomial time (with respect to their sizes) by a deterministic classical algorithm. Then there exists a classical deterministic algorithm $\mathcal{C}$ that receives $X_1 \subseteq \mathcal{X}_1, X_2 \subseteq \mathcal{X}_2, \ldots, X_k \subseteq \mathcal{X}_k$ as input, and outputs a number E such that*

$$\left| \sum_{x_1 \in X_1, x_2 \in X_2, \ldots, x_k \in X_k} f(x_1, x_2, \ldots, x_k) - E \right| \leq \varepsilon \cdot \prod_{i=1}^{k} |X_i|.$$

Furthermore, $\mathcal{C}$ runs in $\sum_{i=1}^{k} |X_i| \cdot 2^{C(\mathcal{P})+S_i(\mathcal{P})}$ time.

Proof. For all $x_i \in \mathcal{X}_i$ we first use $\mathcal{A}_{\mathcal{X}_i}$ in Corollary 29 to calculate $\mathcal{A}_{\mathcal{X}_i}(x_i) \in \mathbb{R}^{2^{O(C(\mathcal{P}))}}$, in $\sum_{i=1}^{k} |X_i| \cdot 2^{C(\mathcal{P})+S_i(\mathcal{P})}$ time. Then we directly output

$$\left\langle \sum_{x_1 \in X_1} \mathcal{A}_{\mathcal{X}_1}(x_1), \sum_{x_2 \in X_2} \mathcal{A}_{\mathcal{X}_2}(x_2), \ldots, \sum_{x_k \in X_k} \mathcal{A}_{\mathcal{X}_k}(x_k) \right\rangle.$$

The correctness simply follows from the fact that for all $(x_1, x_2, \ldots, x_k) \in \prod_i \mathcal{X}_i$,

$$-\varepsilon \leq \langle \mathcal{A}_{\mathcal{X}_1}(x_1), \mathcal{A}_{\mathcal{X}_2}(x_2), \ldots, \mathcal{A}_{\mathcal{X}_k}(x_k) \rangle - f(x_1, x_2, \ldots, x_k) \leq \varepsilon. \qquad \blacktriangleleft$$

▶ Remark. The algorithm described above is actually a sketching algorithm. We may define the sketch for X_i as $\mathsf{sk}_i(X_i) = \sum_{x_i \in X_i} \mathcal{A}_{\mathcal{X}_i}(x_i) \in \mathbb{R}^{2^{O(C(\mathcal{P}))}}$ and the number E can be computed from these sk_i's. This sketching algorithm satisfies a nice additive property, i.e., the sketch of $A \sqcup B$ (union as a multi-set) is simply $\mathsf{sk}_i(A) + \mathsf{sk}_i(B)$.

Now we give approximate counting algorithms for concrete problems, using Theorem 30.

3.1 Counting the k-Tuples of Orthogonal Vectors

The goal of this section is to prove the following theorem.

Reminder of Theorem 4. *For any integers k, d and any constant $\varepsilon > 0$, #k-OV$_{n,d}$ can be approximated deterministically with additive error $\varepsilon \cdot n^k$ in $n \cdot 2^{O(k\sqrt{d})}$ time. In particular, it runs in $n^{1+o(1)}$ time k is a constant and $d = o(\log^2 n)$.*

We first consider quantum communication protocols for the following function f.

▶ **Definition 31.** Let $\mathcal{X}_1 = \mathcal{X}_2 = \ldots = \mathcal{X}_k = \{0,1\}^d$ and

$$f(x_1, x_2, \ldots, x_k) = \begin{cases} 1 & \text{if } \langle x_1, x_2, \ldots, x_k \rangle = 0 \\ 0 & \text{otherwise} \end{cases}.$$

The corresponding communication problem can be solved using the quantum communication protocol in [1] with communication complexity $O(k\sqrt{d})$ and space complexity $O(\text{polylog}(d))$, with constant failure probability. If we use the algorithm in Theorem 30, together with the efficient quantum communication protocol mentioned above, we can then deterministically count the number of k-tuples of orthogonal vectors, in time $n \cdot 2^{O(k \cdot \sqrt{d})}$ time, with an additive $\varepsilon \cdot n^k$ error.

3.2 Counting the Pairs of Orthogonal Sparse Vectors

The goal of this section is to prove the following theorem.

Reminder of Theorem 5. *For integers n, m, d and any constant $\varepsilon > 0$, #Sparse-OV$_{n,m,d}$ can be approximated deterministically with additive error $\varepsilon \cdot n^2$ in*

$$n \cdot 2^{O(d^{2/3}\log(m))}$$

time. In particular, when $m = \mathrm{poly}(d)$ and $d = o\left(\left(\frac{\log n}{\log\log n}\right)^{1.5}\right)$, it runs in $n^{1+o(1)}$ time.

Again we consider quantum communication protocols for the following function f.

▶ **Definition 32.** Let $\mathcal{X} = \mathcal{Y} = \{0,1\}^m_{\leq d}$ and

$$f(x,y) = \begin{cases} 1 & \text{if } \langle x, y\rangle = 0 \\ 0 & \text{otherwise} \end{cases}.$$

The corresponding communication problem can be solved with communication complexity $O(d^{2/3}\log m)$, by simulating the quantum query algorithm in Theorem 25 for Element Distinctness. Too see the connection, let $S = \{i \mid x_i = 1\}$ and $T = \{i \mid y_i = 1\}$. We will have $f(x,y) = 1$ if and only if all elements in $S \sqcup T$ (union as a multi-set) are distinct. Now, using the algorithm in Theorem 30, together with the efficient quantum communication protocol mentioned above, we can deterministically count the number of orthogonal pairs in S and T, in $n \cdot 2^{O(d^{2/3}\log(m))}$ time, with an additive $\varepsilon \cdot n^k$ error.

3.3 Counting Solutions to Formula ∘ SYM Circuits

The goal of this section is to solve the following problem.

Reminder of Problem 1. *Given k sets of n vectors $S_1, S_2, \ldots, S_k \subseteq \{0,\ldots,r\}^d$ and d functions $f_1, f_2, \ldots, f_d$ where each f_i is from $\{0,\ldots,r\}^k$ to $\{0,1\}$, and a Boolean formula $\mathcal{F} : \{0,1\}^d \to \{0,1\}$ of $O(1)$ fan-in. Count the number of k-tuples $u_1 \in S_1, u_2 \in S_2, \ldots, u_k \in S_k$ such that*

$$\mathcal{F}(f_1(u_{1,1}, u_{2,1}, \ldots, u_{k,1}), f_2(u_{1,2}, u_{2,2}, \ldots, u_{k,2}), \ldots, f_d(u_{1,d}, u_{2,d}, \ldots, u_{k,d})) = 1.$$

Reminder of Theorem 7. *For any constant $\varepsilon > 0$, the above problem can be solved deterministically in $n \cdot 2^{O(d^{1/2+o(1)}\cdot k(\log d + \log r))}$ time, within $\varepsilon \cdot n^k$ additive error.*

The corresponding k-party communication problem can be solved by a quantum communication protocol with communication complexity $O(d^{1/2+o(1)} \cdot k(\log d + \log r))$, by simulating the quantum query algorithm for Formula-Evaluation in Theorem 23. For details see Example 27. By our framework, this implies an approximate counting algorithm to the problem mentioned above in time $n \cdot 2^{O(d^{1/2+o(1)}\cdot k(\log d+\log r))}$, with an additive $\varepsilon \cdot n^k$ error.

Here we mention one application to the approximate counting algorithm above.

Reminder of Theorem 6. *For any constant $\varepsilon > 0$, the number of solutions to a Formula ∘ SYM circuit of size m can be approximated deterministically within $\varepsilon \cdot 2^n$ additive error in $2^{O(n^{1/2}m^{1/4+o(1)}\sqrt{\log n + \log m})}$ time. In particular, when $m = n^{2-\delta}$ for some $\delta > 0$, the running time is $2^{o(n)}$.*

Proof of Theorem 6. Consider a $\mathsf{Formula} \circ \mathsf{SYM}$ circuit $\mathcal{C} : \{0,1\}^n \to \{0,1\}$ with m symmetric gates $X_1, X_2, \ldots, X_m$ and a Boolean formula $\mathcal{F}$ of $O(1)$ fan-in. Here we slightly abuse of notation by regarding X_i as a function that maps the number of inputs bits with value one to an output in $\{0,1\}$. We can approximately count the number of solutions to $\mathcal{C}$ as follows.

We split the n inputs bits into s groups, each with n/s input bits. Then for each group, we enumerate all the $2^{n/s}$ possible assignments to the n/s input bits. We create a vector in $\{0, \ldots, n/s\}^m$ for each possible assignment, where the i-th entry is simply the number of ones in the assignment which is an input bit to the i-th symmetric gate X_i. Now, the number of solutions to the circuit $\mathcal{C}$, is simply the same as Problem 1, by setting

$$f_i(u_{1,i}, u_{2,i}, \ldots, u_{k,i}) = X_i(u_{1,i} + u_{2,i} + \ldots + u_{k,i}).$$

The total time complexity would be $2^{n/s} \cdot 2^{O\left(m^{1/2+o(1)} \cdot s(\log m + \log(n/s))\right)}$, with an additive $\varepsilon \cdot 2^n$ error. Setting $s = \frac{n^{1/2}}{m^{1/4+o(1)}\sqrt{\log n + \log m}}$, the final time complexity would be

$$2^{O(n^{1/2} m^{1/4+o(1)} \sqrt{\log n + \log m})}.$$ ◀

4 Algorithms from Arthur-Merlin Communication Protocols

In this section, we prove our algorithmic applications of $\mathsf{AM}^{\mathsf{cc}}$ protocols. We first show faster $\mathsf{AM}^{\mathsf{cc}}$ protocols for F imply faster F-Satisfying-Pair algorithms.

Reminder of Theorem 9. *Let $F : \mathcal{X} \times \mathcal{Y} \to \{0, 1, \perp\}$ be a partial function. Suppose there is a computationally efficient $\mathsf{AM}^{\mathsf{cc}}$ protocol for F with communication complexity T and error ε. Then for n such that $2^T \leq (\sqrt{\varepsilon}n)^{0.1}$, there is an $O\left(\varepsilon n^2 \cdot \mathrm{polylog}(n) + n \cdot 2^T\right)$ time randomized algorithm for F-Satisfying-Pair$_n$.*

Proof. We first assume $n < \frac{1}{10\sqrt{\varepsilon}}$. After drawing a random challenge, for each element $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ we construct a Boolean vector $\mathcal{A}_{\mathcal{X}}(x)$ and $\mathcal{A}_{\mathcal{Y}}(y)$ of length 2^T, where each the i-th entry indicates whether Alice (Bob) accepts when receiving the proof i from Merlin. Here we regard i as a Boolean string of length T via a natural bijection between $[2^T]$ and $\{0,1\}^T$.

According to the guarantee of an $\mathsf{AM}^{\mathsf{cc}}$ protocol, for each $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, when $F(x,y) = 1$, with probability at least $1 - \varepsilon$ over the random challenge, we have $\langle \mathcal{A}_{\mathcal{X}}(x), \mathcal{A}_{\mathcal{Y}}(y) \rangle > 0$, and when $F(a,b) = 0$ we have $\langle \mathcal{A}_{\mathcal{X}}(x), \mathcal{A}_{\mathcal{Y}}(y) \rangle > 0$ with probability at most ε over the random challenge.

By a union bound on all pairs of elements in A and B, we have with probability at least 0.99, for all $a \in A$ and $b \in B$, $\langle \mathcal{A}_A(a), \mathcal{A}_B(b) \rangle > 0$ if and only if $F(a,b) = 1$. Consequently, with probability at least 0.99,

$$\left\langle \sum_{a \in A} \mathcal{A}_A(a), \sum_{b \in B} \mathcal{A}_B(b) \right\rangle > 0$$

if and only if there exist $a \in A$ and $b \in B$ such that $F(a,b) = 1$.

For general $n = |A| = |B|$, we first split A and B into $O(\sqrt{\varepsilon}n)$ groups, each with at most $\frac{1}{10\sqrt{\varepsilon}}$ elements. I.e., we assume $A = \bigcup_{i=1}^{g} A_i$ and $B = \bigcup_{i=1}^{g} B_i$ such that $g = O(\sqrt{\varepsilon}n)$ and $|A_i|, |B_i| \leq \frac{1}{10\sqrt{\varepsilon}}$. For each $i, j \in [g]$, we use the algorithm mentioned above to calculate two vectors $\sum_{a \in A_i} \mathcal{A}_A(a)$ and $\sum_{b \in B_j} \mathcal{A}_B(b)$. We write $\mathcal{M}_A \in \mathbb{R}^{2^T \times g}$ to denote the matrix

$$\left[\sum_{a \in A_1} \mathcal{A}_A(a), \sum_{a \in A_2} \mathcal{A}_A(a), \cdots, \sum_{a \in A_g} \mathcal{A}_A(a) \right]$$

and $\mathcal{M}_B \in \mathbb{R}^{2^T \times g}$ to denote the matrix

$$\left[\sum_{b\in B_1} \mathcal{A}_B(b), \sum_{b\in B_2} \mathcal{A}_B(b), \cdots, \sum_{b\in B_g} \mathcal{A}_B(b)\right].$$

Since $2^T \le (\sqrt{\varepsilon}n)^{0.1} \le O(g^{0.1})$, we can use the rectangular matrix multiplication algorithm in Theorem 16 to calculate $\mathcal{M}_A^T\mathcal{M}_B$ in $O(g^2 \cdot \mathrm{polylog}(g)) = O(\varepsilon n^2 \,\mathrm{polylog}(n))$ time. We repeat this procedure for $O(\log n)$ times. For any $i, j \in [g]$, by standard concentration bounds, with probability at least $1 - \mathrm{poly}(n)$, there exist $a \in A_i$ and $b \in B_j$ such that $F(a, b) = 1$ if and only if the majority of the $O(\log n)$ repetitions satisfies $(\mathcal{M}_A^T\mathcal{M}_B)_{i,j} > 0$. Applying union bound again over all $i, j \in [g]$, we can now solve F-Satisfying-Pair$_n$ by checking whether there exist i and j such that the majority of the $O(\log n)$ repetitions satisfies $(\mathcal{M}_A^T\mathcal{M}_B)_{i,j} > 0$. The overall algorithm runs in $O(\varepsilon n^2 \cdot \mathrm{polylog}(n))$ time and succeeds with high probability, as stated. ◀

4.1 A New Algorithm for Approximate Max-IP

The first application of Theorem 9 is to use the Goldwasser-Sisper AM protocol [28] for approximating set size to obtain a new algorithm for approximating Max-IP.

We first need the following adaption of [28], which has a better dependence on ε.

Reminder of Lemma 11. *There is an* $\mathsf{AM}^{\mathsf{cc}}$ *protocol for* Gap-Inner-Product$_d$ *with error* ε *and communication complexity* $\log\binom{d}{\le O(\log \varepsilon^{-1})}$.

Proof. Recall that $x, y \in \{0,1\}^d$ are the inputs hold by Alice and Bob respectively.

Let $X = \{i \mid x_i = 1\}$ and $Y = \{i \mid y_i = 1\}$. The problem is equivalent to determine whether $|X \cap Y| \ge 2\tau$ or $|X \cap Y| \le \tau$. Here we give an $\mathsf{AM}^{\mathsf{cc}}$ communication protocol with error ε and communication complexity $\log\binom{d}{\le O(\log \varepsilon^{-1})}$.

In the communication protocol, Alice and Bob first generate i.i.d. random variables $p_i \sim \mathrm{Pois}(k/\tau)$ for each $i \in [d]$, for a parameter $k = \Theta(\log(1/\varepsilon))$ to be determined later. When $|X \cap Y| \ge 2\tau$, Merlin finds an arbitrary set $S \subseteq X \cap Y$ of size $O(k)$ such that $\sum_{i\in S} p_i \ge 1.6k$, and then sends it to Alice and Bob. Upon receiving S, Alice (Bob) decides to accept or reject by checking whether $S \subseteq X$ ($S \subseteq Y$) and $\sum_{i\in S} p_i \ge 1.6k$. The communication complexity of this protocol is upper bounded by $\log\binom{d}{\le O(\log \varepsilon^{-1})}$ since $|S| \le 1.6k = O(\log(1/\varepsilon))$.

Now we prove the correctness by considering the following two cases.

Case 1: $|X \cap Y| \ge 2\tau$. For this case, we have $\sum_{i\in X\cap Y} p_i \sim \mathrm{Pois}(|X\cap Y| \cdot k/\tau) \succeq \mathrm{Pois}(2k)$. Thus by Lemma 18, with probability at least $1 - e^{\Omega(k)}$, $\sum_{i\in X\cap Y} p_i \ge 1.6k$. Since for each $p_i > 0$ we must have $p_i \ge 1$, with probability at least $1 - e^{\Omega(k)}$, there exists a set $S \subseteq X \cap Y$ of size $O(k)$ such that $\sum_{i\in S} p_i \ge 1.6k$.

Case 2: $|X \cap Y| \le \tau$. For this case, we have $\sum_{i\in X\cap Y} p_i \sim \mathrm{Pois}(|X \cap Y| \cdot k/\tau) \preceq \mathrm{Pois}(k)$. Thus by Lemma 18, with probability at least $1 - e^{\Omega(k)}$, $\sum_{i\in X\cap Y} p_i \le 1.2k$. When both Alice and Bob accept, it must be the case that $S \subseteq X \cap Y$ and $\sum_{i\in S} p_i \ge 1.6k$. However when $|X \cap Y| \le \tau$, with probability at least $1 - e^{\Omega(k)}$, $\sum_{i\in X\cap Y} p_i \le 1.2k$. Thus there is no S such that both Alice and Bob accept, with probability at least $1 - e^{-\Omega(k)}$.

The lemma follows by setting k to be a large enough multiple of $\log(1/\varepsilon)$. ◀

By Theorem 9 and the above lemma, Corollary 12 follows from a binary search over τ.

Reminder of Corollary 12. *There is an algorithm for computing a* 2*-approximation to* Max-IP$_{n,c\log n}$*, which runs in* $n^{2-1/O(\log c)}$ *time.*

4.2 Consequence of Fast $\mathsf{AM}^{\mathsf{cc}}$ Protocols for LCS and Edit-Distance

Next we discuss the consequences of LCS and Edit-Distance having efficient $\mathsf{AM}^{\mathsf{cc}}$ protocols. We first introduce some classical notations about the communication complexity classes (see [17, 30]). We say a function family $F = \{F_d : \{0,1\}^d \times \{0,1\}^d \to \{0,1,\perp\}\}_{d\in\mathbb{N}}$ is in $\mathsf{AM}^{\mathsf{cc}}$ if $\mathsf{AM}^{\mathsf{cc}}(F_d) = \mathrm{polylog}(d)$ (we use $\mathsf{AM}^{\mathsf{cc}}(F_d)$ to denote the $\mathsf{AM}^{\mathsf{cc}}$ communication complexity for F_d with error $1/3$).

We also say F is $\mathsf{AM}^{\mathsf{cc}}_{\mathsf{eff}}$ if for all $d \in \mathbb{N}$, F_d admits a computationally efficient $\mathsf{AM}^{\mathsf{cc}}$ protocol with error $1/3$ and complexity $\mathrm{polylog}(d)$.

Now we prove the consequence of a function family $F \in \mathsf{AM}^{\mathsf{cc}}_{\mathsf{eff}}$.

▶ **Corollary 33** (Consequence of $F \in \mathsf{AM}^{\mathsf{cc}}_{\mathsf{eff}}$)**.** *Let $F = \{F_d : \{0,1\}^d \times \{0,1\}^d \to \{0,1,\perp\}\}_{d\in\mathbb{N}}$ be a partial function family. If $F \in \mathsf{AM}^{\mathsf{cc}}_{\mathsf{eff}}$, then there is an $n^2/2^{\log^{1-\delta} n}$ time algorithm for $F_{\mathrm{polylog}(n)}$-Satisfying-Pair$_n$, for any constant $\delta > 0$.*

Proof. By standard repetition arguments, there exists an $\mathsf{AM}^{\mathsf{cc}}$ communication protocol with communication complexity $\mathrm{polylog}(d)\log(1/\varepsilon)$ and failure probability $1-\varepsilon$. In order to invoke Theorem 9 we need to make sure

$$2^{\mathrm{polylog}(d)\log(1/\varepsilon)} = 2^{\mathrm{polyloglog}(n)\log(1/\varepsilon)} < n^{0.1},$$

and thus we can set $\varepsilon = 2^{-\log^{1-\delta/2} n}$. For this choice of ε we will then get an $n^2/2^{\log^{1-\delta/2} n} \cdot \mathrm{polylog}(n) \le n^2/2^{\log^{1-\delta} n}$ time algorithm for $F_{\mathrm{polylog}(n)}$-Satisfying-Pair$_n$, which completes the proof. ◀

Recall that in $\mathsf{LCS}^{\mathsf{cc}}_d$ ($\mathsf{Edit\text{-}Dist}^{\mathsf{cc}}_d$), Alice and Bob hold strings $x, y \in \{0,1\}^d$ respectively, and are given an integer τ. Their goal is to decide whether $\mathsf{LCS}(x,y)$ is at least τ ($\mathsf{Edit\text{-}Distance}(x,y)$ is at least τ). Now we are ready to prove Theorem 13.

Reminder of Theorem 13. *If $\mathsf{LCS}^{\mathsf{cc}}_d$ admits computationally efficient $\mathsf{AM}^{\mathsf{cc}}$ protocols with complexity $\mathrm{polylog}(d)$, then Formula-SAT of polynomial-size formulas admits an $2^{n-n^{1-\delta}}$ time algorithm for any constant $\delta > 0$. The same holds for $\mathsf{Edit\text{-}Dist}^{\mathsf{cc}}$ in place of $\mathsf{LCS}^{\mathsf{cc}}$.*

We will only discuss $\mathsf{LCS}^{\mathsf{cc}}$ here, the proof for $\mathsf{Edit\text{-}Dist}^{\mathsf{cc}}$ follows exactly the same. We first introduce the reduction from [3] (see also [2]).

▶ **Theorem 34** (Implicit in [3])**.** *For a given formula $\mathcal{F}$ with n input variables and size s, let $a \in \{0,1\}^{n/2}$ be an assignment to first $n/2$ variables in $\mathcal{F}$ and $b \in \{0,1\}^{n/2}$ be an assignment to last $n/2$ variables in $\mathcal{F}$. There exists an algorithm $\mathcal{A}$ which outputs $G(a) \in \{0,1\}^{\mathrm{poly}(s)}$ and $\overline{G}(b) \in \{0,1\}^{\mathrm{poly}(s)}$ such that for a fixed integer Y (Y depends on $\mathcal{F}$),*

- *$\mathsf{LCS}(G(a), \overline{G}(b)) = Y$ if $a \odot b$ is a satisfying assignment to $\mathcal{F}$;*
- *$\mathsf{LCS}(G(a), \overline{G}(b)) \le Y - 1$ if $a \odot b$ is not a satisfying assignment to $\mathcal{F}$.*

Proof of Theorem 13. For a given formula $\mathcal{F}$ of size $s = \mathrm{poly}(n)$, we first enumerate all $2^{n/2}$ possible assignments to first $n/2$ variables in $\mathcal{F}$ and all possible assignments to last $n/2$ variables in $\mathcal{F}$. For each $a \in \{0,1\}^{n/2}$ corresponding to an assignment to first $n/2$ variables in $\mathcal{F}$ and $b \in \{0,1\}^{n/2}$ corresponding to an assignment to last $n/2$ variables in $\mathcal{F}$, we calculate $G(a)$ and $\overline{G}(b)$ using Theorem 34. Note that all $G(a)$'s and $\overline{G}(b)$'s have length $\mathrm{poly}(s) = \mathrm{poly}(n)$.

Now suppose $\mathsf{LCS}^{\mathsf{cc}} \in \mathsf{AM}^{\mathsf{cc}}_{\mathsf{eff}}$ for $\tau = Y$. Applying Corollary 33 with all possible $G(a)$'s and $\overline{G}(b)$'s, we can solve Formula-SAT in $2^{n-n^{1-\delta}}$ time for any constant $\delta > 0$. ◀

Open Problems and Future Directions

Here we list a few interesting open problems stemming from this work.

- In this work, we applied $\mathsf{BQP}^{\mathsf{cc}}$ and $\mathsf{AM}^{\mathsf{cc}}$ protocols for the algorithmic purpose. Can we find algorithmic applications of other communication protocols?
- Or less ambitiously, can we find more interesting algorithmic applications with other known $\mathsf{BQP}^{\mathsf{cc}}$ or $\mathsf{AM}^{\mathsf{cc}}$ protocols? (this could even be a motivation to find *new* $\mathsf{BQP}^{\mathsf{cc}}$ or $\mathsf{AM}^{\mathsf{cc}}$ protocols!)
- Our additive approximation algorithm for #OV runs in near-linear time when $d = o(\log^2 n)$. Is it possible to design a near-linear time algorithm for $d = n^{o(1)}$ dimensions? Note that by a simple Chernoff bound, there is a deterministic $n^{1+o(1)}$ time algorithm with $n^{1+o(1)}$ advice for additive approximations to #OV. So there is a hope to construct such an algorithm.
- Our results show that under the hypothesis of [2], $\mathsf{LCS}^{\mathsf{cc}}$ and $\mathsf{Edit\text{-}Dist}^{\mathsf{cc}}$ do not admit computationally efficient $\mathsf{AM}^{\mathsf{cc}}$ or $\mathsf{PH}^{\mathsf{cc}}$ protocols. Can one prove that *unconditionally*?
- Is it possible to connect these algorithms from $\mathsf{AM}^{\mathsf{cc}}$ or $\mathsf{PH}^{\mathsf{cc}}$ protocols to R. Williams' algorithmic approach to circuit lower bounds [49, 51, 38]? In particular, can one show *unconditionally* that, there is a function f in NEXP (or even $\mathsf{NTIME}[2^{\mathrm{polylog}(n)}]$), which doesn't admit polylog(n) complexity $\mathsf{AM}^{\mathsf{cc}}$ or $\mathsf{PH}^{\mathsf{cc}}$ protocols?

References

1. Scott Aaronson and Andris Ambainis. Quantum Search of Spatial Regions. *Theory of Computing*, 1(1):47–79, 2005.
2. Amir Abboud and Karl Bringmann. Tighter Connections Between Formula-SAT and Shaving Logs. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, pages 8:1–8:18, 2018.
3. Amir Abboud, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Ryan Williams. Simulating branching programs with edit distance and friends: or: a polylog shaved is a lower bound made. In *Proc. of the 48th STOC*, pages 375–388, 2016.
4. Amir Abboud and Aviad Rubinstein. Fast and Deterministic Constant Factor Approximation Algorithms for LCS Imply New Circuit Lower Bounds. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 35:1–35:14, 2018.
5. Amir Abboud, Aviad Rubinstein, and R. Ryan Williams. Distributed PCP Theorems for Hardness of Approximation in P. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 25–36, 2017.
6. Amir Abboud, Richard Ryan Williams, and Huacheng Yu. More Applications of the Polynomial Method to Algorithm Design. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 218–230, 2015.
7. Josh Alman. An Illuminating Algorithm for the Light Bulb Problem. In *SOSA*, 2019.
8. Josh Alman, Timothy M. Chan, and R. Ryan Williams. Polynomial Representations of Threshold Functions and Algorithmic Applications. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 467–476, 2016.
9. Josh Alman and R. Ryan Williams. Probabilistic rank and matrix rigidity. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 641–652, 2017.

10 Josh Alman and Ryan Williams. Probabilistic Polynomials and Hamming Nearest Neighbors. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 136–150, 2015.

11 Noga Alon. Perturbed identity matrices have high rank: Proof and applications. *Combinatorics, Probability and Computing*, 18(1-2):3–15, 2009.

12 Noga Alon, Troy Lee, and Adi Shraibman. The cover number of a matrix and its algorithmic applications. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 28. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

13 Noga Alon, Troy Lee, Adi Shraibman, and Santosh Vempala. The approximate rank of a matrix and its algorithmic applications: approximate rank. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 675–684. ACM, 2013.

14 Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007.

15 Andris Ambainis, Andrew M Childs, Ben W Reichardt, Robert Špalek, and Shengyu Zhang. Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. *SIAM Journal on Computing*, 39(6):2513–2530, 2010.

16 Andris Ambainis, Leonard J. Schulman, Amnon Ta-Shma, Umesh V. Vazirani, and Avi Wigderson. The Quantum Communication Complexity of Sampling. *SIAM J. Comput.*, 32(6):1570–1585, 2003.

17 László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory (preliminary version). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 337–347, 1986.

18 Boaz Barak, Zeev Dvir, Amir Yehudayoff, and Avi Wigderson. Rank bounds for design matrices with applications to combinatorial geometry and locally correctable codes. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 519–528. ACM, 2011.

19 Harry Buhrman, Richard Cleve, and Avi Wigderson. Quantum vs. Classical Communication and Computation. In *Proc. of the Thirtieth Annual ACM Symposium on the Theory of Computing*, pages 63–68, 1998.

20 Harry Buhrman, Richard Cleve, and Avi Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 63–68. ACM, 1998.

21 Harry Buhrman and Ronald de Wolf. Communication complexity lower bounds by polynomials. In *Computational Complexity, 16th Annual IEEE Conference on, 2001.*, pages 120–130. IEEE, 2001.

22 Timothy M. Chan and Ryan Williams. Deterministic APSP, Orthogonal Vectors, and More: Quickly Derandomizing Razborov-Smolensky. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1246–1255, 2016.

23 Lijie Chen. On The Hardness of Approximate and Exact (Bichromatic) Maximum Inner Product. In *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, pages 14:1–14:45, 2018.

24 Lijie Chen, Shafi Goldwasser, Kaifeng Lyu, Guy Rothblum, and Aviad Rubinstein. Fine-grained Complexity Meets IP = PSPACE. In *SODA*, 2019.

25 Lijie Chen and Ryan Williams. An Equivalence Class for Orthogonal Vectors. In *SODA*, 2019.

26 Don Coppersmith. Rapid multiplication of rectangular matrices. *SIAM Journal on Computing*, 11(3):467–471, 1982.

27 Francois Le Gall and Florent Urrutia. Improved rectangular matrix multiplication using powers of the Coppersmith-Winograd tensor. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1029–1046. SIAM, 2018.

28 Shafi Goldwasser and Michael Sipser. Private Coins versus Public Coins in Interactive Proof Systems. *Advances in Computing Research*, 5:73–90, 1989.

29 Mika Göös, Toniann Pitassi, and Thomas Watson. Zero-Information Protocols and Unambiguity in Arthur-Merlin Communication. *Algorithmica*, 76(3):684–719, 2016.

30 Mika Göös, Toniann Pitassi, and Thomas Watson. The Landscape of Communication Complexity Classes. *Computational Complexity*, 27(2):245–304, 2018.

31 Parikshit Gopalan, Ryan O'Donnell, Yi Wu, and David Zuckerman. Fooling Functions of Halfspaces under Product Distributions. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, USA, June 9-12, 2010*, pages 223–234, 2010.

32 Prahladh Harsha, Adam R. Klivans, and Raghu Meka. An invariance principle for polytopes. *J. ACM*, 59(6):29:1–29:25, 2012.

33 Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1283–1296, 2018.

34 Ilan Kremer. *Quantum communication*. Citeseer, 1995.

35 Troy Lee and Adi Shraibman. Lower Bounds in Communication Complexity. *Foundations and Trends in Theoretical Computer Science*, 3(4):263–398, 2009.

36 Daniel Lokshtanov, Ramamohan Paturi, Suguru Tamaki, R. Ryan Williams, and Huacheng Yu. Beating Brute Force for Systems of Polynomial Equations over Finite Fields. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2190–2202, 2017.

37 Kurt Mehlhorn and Erik Meineche Schmidt. Las Vegas Is better than Determinism in VLSI and Distributed Computing (Extended Abstract). In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 330–337, 1982.

38 Cody Murray and R. Ryan Williams. Circuit lower bounds for nondeterministic quasi-polytime: an easy witness lemma for NP and NQP. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 890–901, 2018.

39 Ryan O'Donnell, Rocco A. Servedio, and Li-Yang Tan. Fooling Polytopes. *CoRR*, abs/1808.04035, 2018.

40 Ramamohan Paturi and Janos Simon. Probabilistic Communication Complexity. *J. Comput. Syst. Sci.*, 33(1):106–123, 1986.

41 Alexander A Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987.

42 Ben Reichardt. Reflections for quantum query algorithms. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 560–569, 2011.

43 Karthik C. S. and Pasin Manurangsi. On Closest Pair in Euclidean Metric: Monochromatic is as Hard as Bichromatic. In *ITCS*, 2019.

44 Rocco A. Servedio and Li-Yang Tan. Fooling Intersections of Low-Weight Halfspaces. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 824–835, 2017.

45 Roman Smolensky. Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 77–82, 1987.

46 Zhao Song, David P Woodruff, and Peilin Zhong. Relative Error Tensor Low Rank Approximation. In *SODA*, 2019.

47 Avishay Tal. #SAT algorithms from shrinkage. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:114, 2015.

48 Richard Ryan Williams. The Polynomial Method in Circuit Complexity Applied to Algorithm Design (Invited Talk). In *34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, December 15-17, 2014, New Delhi, India*, pages 47–60, 2014.

49 Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM Journal on Computing*, 42(3):1218–1244, 2013.

50 Ryan Williams. Faster all-pairs shortest paths via circuit complexity. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 664–673. ACM, 2014.

51 Ryan Williams. Non-Uniform ACC circuit lower bounds. *Journal of the ACM (JACM)*, 61(1):2, 2014.

52 Huacheng Yu. Optimal Succinct Rank Data Structure via Approximate Nonnegative Tensor Decomposition. *arXiv preprint*, 2018. `arXiv:1811.02078`.